

# 7

## Diskrete und schnelle FT

### 7.1 Algebra

Die diskrete Fourier-Transformation, abgekürzt **DFT**, kommt unter anderem in den folgenden Zusammenhängen zum Zug:

- (a) numerische Fourier-Analyse von  $L$ -periodischen Funktionen  $f: \mathbb{R} \rightarrow \mathbb{C}$ ,
- (b) numerische Fourier-Analyse von Zeitsignalen  $f: \mathbb{R} \rightarrow \mathbb{C}$ , die ausserhalb eines Intervalls der Länge  $L$  identisch verschwinden,
- (c) numerische Fourier-Analyse von Ausschnitten  $f \upharpoonright [0, L]$  aus Zeitsignalen oder von Bildausschnitten,
- (d) Fourier-Analyse von  $N$ -periodischen Funktionen  $y: \mathbb{Z} \rightarrow \mathbb{C}$ .

In allen Fällen haben wir  $N \gg 1$  äquidistante Stützstellen, und in allen Fällen ist der resultierende Rechenprozess derselbe. Am klarsten ist die Situation im Fall (d). Hier ist die DFT exakt; es gibt keine Konvergenzfragen, und während des Rechenprozesses geht keine Information verloren: Die Ausgangsfunktion  $f$  lässt sich aus ihrer Fourier-Transformierten  $\hat{f}$  (auch sie  $N$ -periodisch) eins-zu-eins zurückgewinnen. Bei der Abbildung der in der Praxis wichtigen Fälle (a)–(c) auf (d) treten hingegen typische Diskretisierungsfehler, Aliasing und Ähnliches auf, die besonders studiert werden müssen. Auf diesen Punkt werden wir im nächsten Abschnitt eingehen.

Wir befassen uns also zunächst mit (d). Es sei  $N \gg 1$  gegeben. Eine  $N$ -periodische Funktion  $y: \mathbb{Z} \rightarrow \mathbb{C}$  ist vollständig bestimmt durch ihre Werte auf der Menge  $[N] := \{0, 1, 2, \dots, N-1\}$  und kann daher mit dem Datenvektor

$$\mathbf{y} := (y_0, y_1, \dots, y_{N-1}), \quad y_k := y(k) \quad (0 \leq k \leq N-1)$$

identifiziert werden. Für die Zwecke der Matrizenrechnung betrachten wir  $\mathbf{y}$  als einen *Kolonnenvektor*. Die Gesamtheit  $X$  dieser Vektoren ist ein  $N$ -dimensionaler komplexer Vektorraum mit dem Skalarprodukt

$$\langle \mathbf{x}, \mathbf{y} \rangle := \sum_{k=0}^{N-1} x_k \overline{y_k}.$$

Wir bestimmen zunächst die im Fall (d) massgebenden harmonischen Grundfunktionen

$$\mathbf{e}_\alpha: \mathbb{Z} \rightarrow \mathbb{C}, \quad k \mapsto e^{i\alpha k}.$$

Ein derartiges  $\mathbf{e}_\alpha$  besitzt genau dann die Periode  $N$ , wenn  $e^{i\alpha N} = 1$  ist, und das heisst:  $\alpha = 2n\pi/N =: \alpha_n$  für ein  $n \in \mathbb{Z}$ . Weiter: Der Wert von  $\mathbf{e}_\alpha$  ist auf ganz  $\mathbb{Z}$  bestimmt durch den Wert an der Stelle  $k = 1$ , und zwei Funktionen  $\mathbf{e}_\alpha, \mathbf{e}_{\alpha'}$ , die an der Stelle  $k = 1$  denselben Wert haben, stimmen auf ganz  $\mathbb{Z}$  überein. Wir müssen also untersuchen, unter welchen Umständen

$$\mathbf{e}_{\alpha_n}(1) = \mathbf{e}_{\alpha_m}(1) \quad \text{d.h.} \quad e^{2n\pi i/N} = e^{2m\pi i/N}$$

gilt. Dies trifft genau dann zu, wenn  $e^{2(n-m)\pi i/N} = 1$  ist, und das heisst: wenn sich  $n$  und  $m$  um ein Vielfaches von  $N$  unterscheiden. Damit verbleiben als tatsächlich *verschiedene* harmonische Grundfunktionen auf  $\mathbb{Z}/N$  die  $N$  Funktionen

$$\mathbf{e}_{2n\pi/N}: \mathbb{Z} \rightarrow \mathbb{C}, \quad k \mapsto e^{2nk\pi i/N} \quad (n \in [N]),$$

die wir im weiteren mit  $\chi_n$  bezeichnen ( $\chi$  für *Charakter*), siehe die Fig. 7.1.1.

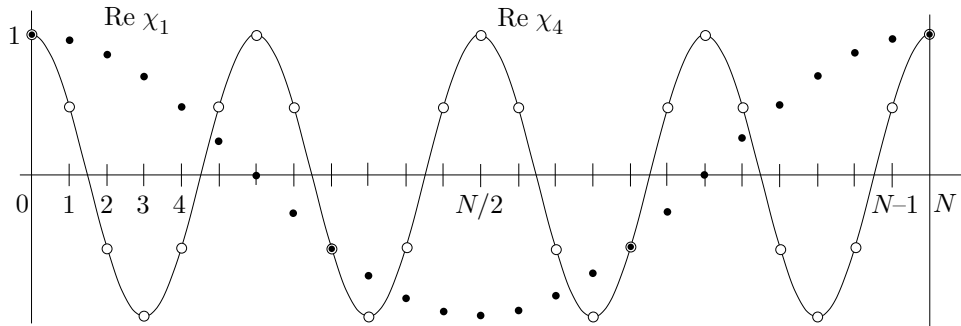


Fig. 7.1.1

Mit der Abkürzung

$$e^{2\pi i/N} =: w_N =: w$$

(eine  $N$ -te Einheitswurzel, siehe die Fig. 7.1.2) lässt sich also  $\chi_n$  folgendermassen schreiben:

$$\chi_n(k) = w^{nk} \quad (k \in \mathbb{Z}, n \in [N]) .$$

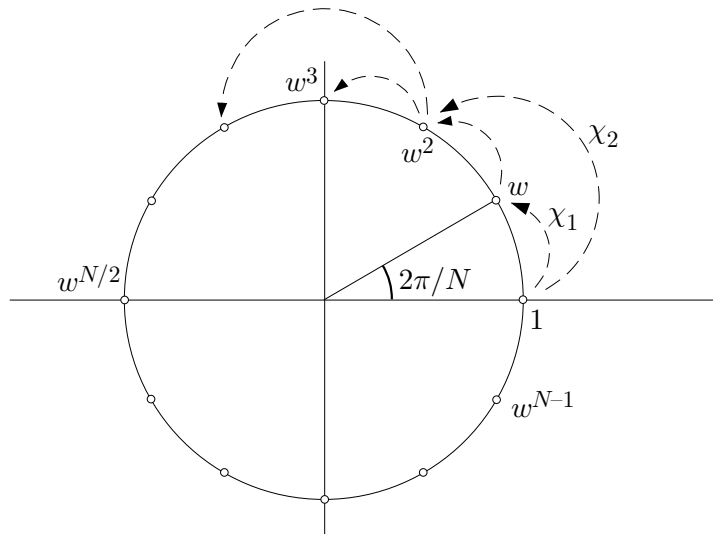


Fig. 7.1.2

**(7.1)** Die  $\chi_n$  ( $n \in [N]$ ) bilden im wesentlichen eine orthonormierte Basis von  $X$ , genauer:

$$\langle \chi_n, \chi_m \rangle = N \delta_{nm} \quad (m, n \in [N]) .$$

□ Im Fall  $m = n$  ist  $\langle \chi_n, \chi_m \rangle = \sum_{k=0}^{N-1} 1 = N$ , und im Fall  $m \neq n$  ist

$$\langle \chi_n, \chi_m \rangle = \sum_{k=0}^{N-1} w^{(n-m)k} = \frac{1 - w^{(n-m)N}}{1 - w^{n-m}} = 0 ,$$

da der Nenner nicht verschwindet. □

Auf Grund allgemeiner Prinzipien der linearen Algebra können wir daher folgendes sagen: Jeder Vektor  $\mathbf{y} \in X$  lässt sich als Linearkombination der  $\chi_n$  darstellen:

$$\mathbf{y} = \sum_{n=0}^{N-1} \hat{y}_n \chi_n, \quad (1)$$

oder ausgeschrieben:

$$(7.2) \quad y_k = \sum_{n=0}^{N-1} \hat{y}_n w^{nk} \quad (0 \leq k \leq N-1).$$

Dabei sind die Koeffizienten  $\hat{y}_n$  gegeben durch

$$(7.3) \quad \hat{y}_n = \frac{1}{N} \langle \mathbf{y}, \chi_n \rangle = \frac{1}{N} \sum_{k=0}^{N-1} y_k w^{-nk} \quad (0 \leq n \leq N-1).$$

In Matrixschreibweise sieht das folgendermassen aus: Man bildet die symmetrische  $(N \times N)$ -Matrix  $\mathbf{W}$  mit Elementen  $w^{nk}$  ( $0 \leq n, k \leq N-1$ ):

$$\mathbf{W} := \mathbf{W}_N := \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & w & w^2 & & w^{N-1} \\ 1 & w^2 & w^4 & & \vdots \\ \vdots & & & & \\ 1 & w^{N-1} & \dots & & w^{(N-1)(N-1)} \end{bmatrix}. \quad (2)$$

Dann sind (7.2) und (7.3) äquivalent mit den Formeln

$$(7.4) \quad \mathbf{y} = \mathbf{W} \hat{\mathbf{y}}, \quad \hat{\mathbf{y}} = \frac{1}{N} \overline{\mathbf{W}} \mathbf{y}.$$

Die vektorielle Gleichung (1) codiert die folgende Aussage über die  $N$ -periodische Funktion  $y: \mathbb{Z} \rightarrow \mathbb{C}$ :

$$y(k) = \sum_{n=0}^{N-1} \hat{y}_n \chi_n(k) \quad (k \in \mathbb{Z}),$$

sie stellt also das  $y$  als Linearkombination der harmonischen Grundfunktionen  $\chi_n$  dar und kann somit als Fourier-Entwicklung von  $y$  aufgefasst werden. Die  $N$  Zahlen  $\hat{y}_n$  ( $n \in [N]$ ) heissen deshalb **Fourier-Koeffizienten** von  $y$ .

Die Berechnung der Fourier-Koeffizienten  $\hat{y}_n$  sowie die Rücktransformation laufen nach (7.4) auf eine einfache Matrizenmultiplikation hinaus. Das sind je etwa  $N^2$  Multiplikationen und ungefähr ebensoviele Additionen. Wir werden in Abschnitt 7.3 sehen, dass die spezielle Gestalt der Matrix  $\mathbf{W}$  erlaubt, den für eine  $N$ -DFT notwendigen Rechenaufwand im wesentlichen auf  $0.5 N \log_2 N$  Multiplikationen und etwa doppelt soviele Additionen herabzusetzen. Dieser Sachverhalt ist der Hauptgrund für die grosse Beliebtheit der DFT in den verschiedensten Anwendungsgebieten.

## 7.2 Diskretisierung

Wir gehen nun daran, die Paradigmen (a)–(c) des vorangehenden Abschnitts mit der in (7.1)–(7.4) beschriebenen “abstrakten” DFT zu verbinden.

(a) Es sei

$$f : \mathbb{R} \rightarrow \mathbb{C}, \quad f(t + L) \equiv f(t),$$

ein  $L$ -periodisches Zeitsignal. Die Funktion  $f$  besitzt eine Fourier-Entwicklung der Form

$$f(t) = \sum_{n=-\infty}^{\infty} c_n e^{2n\pi it/L}, \quad (1)$$

wobei die Koeffizienten  $c_n$  nach (5.3)(e) gegeben sind durch

$$c_n = \frac{1}{L} \int_0^L f(t) e^{-2n\pi it/L} dt \quad (n \in \mathbb{Z}). \quad (2)$$

Dieses Signal wird nun an den  $N$  äquidistanten Stützstellen  $t_k := kL/N$  abgetastet; man erhält den Datenvektor  $\mathbf{y}$  mit den Einträgen

$$y_k := f\left(\frac{k}{N}L\right) \quad (0 \leq k \leq N-1).$$

Wie lassen sich diese  $y_k$  zur approximativen Berechnung der Fourier-Koeffizienten  $c_n$  verwenden? Ersetzen wir das Integral (2) durch die mit den  $y_k$  gebildete "Trapezsumme" (es handelt sich um das Integral der zwischen den Stützstellen  $t_k$  linear interpolierten Funktion), so ergibt sich die Näherung

$$c_n \doteq \frac{1}{L} \sum_{k=0}^{N-1} f(t_k) e^{-2n\pi i t_k / L} \frac{L}{N} .$$

Nun ist

$$e^{-2n\pi i t_k / L} = e^{-2n\pi i k / N} = w^{-nk} , \quad (3)$$

wobei wir wiederum  $e^{2\pi i / N} =: w$  gesetzt haben. Mit Rücksicht auf (7.3) erhalten wir daher

$$c_n \doteq \frac{1}{N} \sum_{k=0}^{N-1} y_k w^{-nk} = \hat{y}_n . \quad (?)$$

In Worten: Die mit DFT berechneten Fourier-Koeffizienten  $\hat{y}_n$  sind Näherungswerte für die effektiven Fourier-Koeffizienten  $c_n$  des betrachteten Zeitsignals. Da kann doch etwas nicht stimmen! Es gibt unendlich viele  $c_n$ , aber nur genau  $N$  Zahlen  $\hat{y}_n$ .

Um zu ergründen, was wirklich passiert, überlegen wir folgendermassen: Nach (1) und (3) ist

$$y_k = f(t_k) = \sum_{l=-\infty}^{\infty} c_l e^{2l\pi i t_k / L} = \sum_{l=-\infty}^{\infty} c_l w^{lk} \quad (0 \leq k \leq N-1) .$$

Schreiben wir hier

$$l := n + Nj \quad (0 \leq n \leq N-1, -\infty < j < \infty) ,$$

so gilt

$$w^{lk} = w^{(n+Nj)k} = w^{nk} ,$$

und es ergibt sich die folgende Darstellung von  $y_k$ :

$$y_k = \sum_{n=0}^{N-1} \left( \sum_{j=-\infty}^{\infty} c_{n+Nj} \right) w^{nk} .$$

Auf Grund von (7.2) haben wir damit definitiv

$$\hat{y}_n = \sum_{j=-\infty}^{\infty} c_{n+jN} \quad (0 \leq n \leq N-1) . \quad (4)$$

Der durch DFT der Daten  $y_k$  erhaltene Näherungswert für  $c_n$  ist also in Wahrheit eine Summe von unendlich vielen Fourier-Koeffizienten. Das hat folgende Konsequenz: Wir können zwar die  $\hat{y}_n$  verwenden, um mit Hilfe von (7.2) die Abtastwerte  $y_k$  exakt zu reproduzieren. Wenn wir aber  $f$  mit Hilfe der  $\hat{y}_n$  für alle reellen  $t$  wiederherstellen wollen, so ist die erhaltene Kopie  $\tilde{f}$  mit Aliasing belastet.

Um dieses Aliasing wegzubedingen, wollen wir jetzt annehmen, unser Zeitsignal  $f$  sei  $(N/2)$ -**bandbegrenzt**; das heisst, es gelte

$$c_n = \hat{f}(n) = 0 \quad \left( |n| \geq \frac{N}{2} \right). \quad (5)$$

In der Praxis wird  $f$  nicht bandbegrenzt sein. Man kann aber folgendes sagen: Je öfter  $f$  differenzierbar ist, desto schneller konvergieren die  $c_n$  mit  $|n| \rightarrow \infty$  gegen 0. Gilt nun (5), so lassen sich die  $\hat{y}_n$  wie folgt identifizieren:

$$\hat{y}_n = \begin{cases} c_n & (0 \leq n < N/2) \\ 0 & (n = N/2) \\ c_{n-N} & (N/2 < n \leq N-1) \end{cases}; \quad (6)$$

denn die Summe (4) enthält für jedes  $n$  höchstens *einen* nichtverschwindenden Term. Wird das System (6) nach den  $c_n$  aufgelöst, so ergibt sich

$$c_n = \begin{cases} \hat{y}_{n+N} & (-N/2 < n < 0) \\ \hat{y}_n & (0 \leq n < N/2) \\ 0 & (|n| \geq N/2) \end{cases}. \quad (7)$$

Wir vereinbaren noch die Schreibweise

$$\hat{y}_n := \hat{y}_{n+jN} \quad (n + jN \in [N]). \quad (8)$$

Tragen wir nun (7) in die Darstellung (1) ein, so ergibt sich das folgende "Shannon-Theorem" für periodische Funktionen:

**(7.5)** *Ist  $f$  eine  $(N/2)$ -bandbegrenzte periodische Funktion, so lässt sie sich aus  $N$  Abtastwerten für alle  $t \in \mathbb{R}$  exakt reproduzieren. Es gilt*

$$f(t) = \sum_{|n| < N/2} \hat{y}_n e^{2n\pi it/L} \quad (t \in \mathbb{R}),$$

wobei die  $\hat{y}_n$  durch (7.2) gegeben sind.

① Nebenbei: Wir mussten  $c_{N/2} = c_{-N/2} = 0$  verlangen. Die beiden Funktionen

$$f_+(t) := e^{N\pi it/L}, \quad f_-(t) := e^{-N\pi it/L}$$

sind verschieden, liefern aber dieselben  $N$  Abtastwerte  $y_k = (-1)^k$ .  $\bigcirc$

Ist  $f$  in Tat und Wahrheit nicht bandbegrenzt, so sind die mit DFT berechneten  $\hat{y}_n$  nur noch Näherungswerte für die  $c_n$  mit  $|n| \leq N/2$ . Wir können aber trotzdem den Ausdruck

$$\tilde{f}(t) := \sum'_{|n| \leq N/2} \hat{y}_n e^{2n\pi it/L} \quad (9)$$

bilden, wobei der  $'$  darauf hinweist, dass die beiden zu  $n = \pm N/2$  gehörenden Terme (so vorhanden) nur mit dem Gewicht  $\frac{1}{2}$  zu berücksichtigen sind. Die Funktion  $\tilde{f}$  ist ein trigonometrisches Polynom vom Grad  $\leq N/2$ , das die gemessenen Werte  $y_k = f(kL/N)$  exakt interpoliert.

(b) Wir werden uns hier etwas kürzer fassen. — Es sei

$$f: \mathbb{R} \rightarrow \mathbb{C}, \quad f(t) \equiv 0 \quad (t \leq 0 \vee t \geq L),$$

ein stetiges Zeitsignal, das ausserhalb des Intervalls  $]0, L[$  identisch verschwindet. Dieses Zeitsignal besitzt eine Fourier-Transformierte

$$\hat{f}(\xi) = \int_0^L f(t) e^{-i\xi t} dt \quad (\xi \in \mathbb{R}). \quad (10)$$

Dieses Signal wird jetzt an den äquidistanten Stützstellen  $t_k := kL/N$  abgetastet. Man erhält wiederum einen Datenvektor  $\mathbf{y}$  mit Einträgen

$$y_k := f\left(\frac{k}{N}L\right) \quad (0 \leq k \leq N-1);$$

dabei fällt automatisch  $y_0 = 0$  aus, und auch das (nicht mehr berücksichtigte)  $y_N$  hätte den Wert 0.

Wir wollen nun diese  $y_k$  zur angenäherten Berechnung der Fourier-Transformierten  $\hat{f}$  verwenden, und zwar berechnen wir  $\hat{f}$  an den (passend gewählten, s.u.) Stellen

$$\xi_n = \frac{2n\pi}{L} \quad \left(-\frac{N}{2} \leq n \leq \frac{N}{2}\right).$$



Ersetzen wir das Integral (10) wieder durch die mit den  $y_k$  gebildete Trapezsumme, so ergibt sich

$$\widehat{f}(\xi_n) \doteq \sum'_{k=0}^N f(t_k) e^{-i\xi_n t_k} \frac{L}{N} = \frac{L}{N} \sum'_{k=0}^N y_k e^{-2nk\pi i/N} = L \frac{1}{N} \sum'_{k=0}^N y_k w^{-nk} .$$

Gemäss der DFT-Grundformel (7.3) haben wir daher

$$\widehat{f}(\xi_n) \doteq L \widehat{y}_n \quad \left( -\frac{N}{2} \leq n \leq \frac{N}{2} \right), \quad (?)$$

wobei wir uns weiterhin an die Konvention (8) halten.

Inwiefern lässt sich dieses Vorgehen rechtfertigen? Hierzu überlegen wir folgendermassen: Die beim Abtasten von  $f$  verwendete Gitterweite hat den Wert  $L/N$ . Zu dieser Gitterweite gehört nach dem Theorem von Shannon (6.4) die maximale erkennbare Frequenz  $\Omega := \pi N/L$ . Genauer: Wäre unser Signal  $\Omega$ -bandbegrenzt, d.h.  $\widehat{f}(\xi) \equiv 0$  für  $|\xi| \geq \Omega$ , so wäre  $f$  (und damit auch  $\widehat{f}$ ) durch den Datenvektor  $\mathbf{y}$  vollständig bestimmt. Nun kann ein Signal mit kompaktem Träger  $[0, L]$  unmöglich bandbegrenzt sein, denn  $\widehat{f}$  ist dann automatisch eine analytische Funktion der komplexen Variablen  $\xi$ .

Die gewählten Punkte  $\xi_n$  bilden gerade eine äquidistante Unterteilung des Intervalls  $[-\Omega, \Omega]$ . Wenn wir nun annehmen, dass  $\widehat{f}(\xi)$  ausserhalb dieses Intervalls vernachlässigbar klein ist, so findet praktisch kein Aliasing statt, und die eben mit einem Fragezeichen (?) versehene Formel lässt sich durchaus als eine Näherungsformel für den Wert  $\widehat{f}(\xi_n)$  ansehen. Die exakte Formel lautet

$$L \widehat{y}_n = \sum_{j=-\infty}^{\infty} \widehat{f}(\xi_n + 2j\Omega),$$

was man ähnlich wie (4) beweist.

Weiter: Aus den  $\widehat{y}_n$  lassen sich die  $y_k$  mit Hilfe von (7.2) exakt zurückgewinnen. In der Folge können wir nun ein trigonometrisches Polynom  $\tilde{f}$  vom Grad  $\leq N/2$  angeben, das  $f$  an den Stellen  $t_k$  exakt interpoliert, nämlich

$$\tilde{f}(t) := \sum'_{|n| \leq N/2} \widehat{y}_n e^{2n\pi i t/L} .$$

Es handelt sich natürlich um dieselbe Funktion wie in (9), mit dem Unterschied, dass nunmehr die Ersatzfunktion  $\tilde{f}$  nicht  $L$ -periodisch, sondern  $\equiv 0$  auf ganz  $\mathbb{R}$  fortgesetzt werden soll.

(c) Erfasst der Datenvektor  $\mathbf{y}$  nur einen Ausschnitt aus einem durchlaufenden Signal  $f$ , so sind die Anfangswerte  $y_0, y_1, y_2$  nicht mit den Endwerten  $y_{N-3}, y_{N-2}, y_{N-1}$  korreliert; in anderen Worten: Die von der Theorie supponierte  $L$ -periodische Funktion  $f^*$  hat für alle praktischen Zwecke eine Sprungstelle, und das hat zur Folge, dass die Fourier-Koeffizienten wesentlich langsamer abnehmen als in den Fällen (a) und (b), mit den bekannten Konsequenzen. Wir gehen auf diesen Punkt nicht weiter ein, um so mehr, als er auch in der Praxis kaum angesprochen wird.

## 7.3 Schnelle Fourier-Transformation

Die sogenannte **schnelle Fourier-Transformation**, abgekürzt **FFT** (für *fast Fourier transform*), ist 1965 von Cooley & Tukey offiziell ins Leben gerufen worden. Es gibt aber Vorläufer; insbesondere findet sie sich schon in einem posthumen *paper* von Gauss (erschienen 1866).

Der eigentliche mathematische Kern der FFT ist eine gewisse Symmetrieüberlegung, die sich in wenigen Zeilen erklären lässt. Dieses Grundprinzip funktioniert, sobald die Länge  $N$  des betrachteten Datensatzes  $\mathbf{y}$  in ein Produkt zerlegt wird:  $N = p \cdot q$ , und stellt die  $N$ -DFT als eine “verschränkte Hintereinanderschaltung” von Transformationen dar, die je nur Datensätze der Länge  $p$  bzw.  $q$  betreffen. Die Anzahl der insgesamt notwendigen Multiplikationen reduziert sich dabei von  $N^2$  auf ungefähr  $N(p + q)$ . Ist zum Beispiel  $p = q$ , so sind das nur noch  $2N^{3/2}$  Multiplikationen. Man wird nun  $p$  und  $q$  ihrerseits in Faktoren zerlegen und so eine weitere Reduktion des Rechenaufwands erzielen, undsoweiter. Alles in allem scheint es, dass sich am meisten herausholen lässt, wenn  $N$  eine Zweierpotenz ist. Im folgenden wollen wir uns auf diesen Fall beschränken.

Die DFT ist mit Hilfe der üblichen Operationen der Analysis und des Summenoperators  $\sum$  erklärt. Im Gegensatz dazu spielt sich die FFT zum grossen Teil in der “Adressverwaltung” ab, und das ist nicht leicht mit der handelsüblichen mathematischen Typographie darzustellen. Man hilft sich dann mit Flussdiagrammen und bildhaften Ausdrücken wie “Schmetterlingsmatrix”.

Es geht also um die Operation

$$\mathbf{x} \mapsto \mathbf{y} := \mathbf{W} \mathbf{x}; \quad (1)$$

dabei ist  $N$  eine gerade Zahl,  $\mathbf{W}$  die DFT-Matrix 7.1.(2), und  $\mathbf{x}$ ,  $\mathbf{y}$  sind Kolonnenvektoren der Länge  $N$ . Wir halten uns an die folgenden Bezeichnungen:

$$w := e^{2\pi i/N}, \quad N' := \frac{N}{2}, \quad w' := e^{2\pi i/N'} = w^2.$$

Weiter bezeichne  $\mathbf{W}'$  die mit  $w'$  gebildete  $(N' \times N')$ -Matrix 7.1.(2). Und noch ein Letztes: Ist  $\mathbf{z}$  ein Kolonnenvektor beliebiger Länge  $M$ , so bezeichnen wir den Vektor  $\mathbf{W}_M \mathbf{z}$  wahlweise auch mit  $\mathbf{z}^{\triangleleft}$ . Beispiel:

$$\begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix}^{\triangleleft} := \mathbf{W}_4 \begin{bmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \end{bmatrix}.$$

Wir können nun beginnen. Die Beziehung (1) lautet in Komponenten ausgeschrieben:  $y_n = \sum_{k=0}^{N-1} w^{nk} x_k$ . Werden hier die geraden und die ungeraden  $k$  voneinander getrennt, so ergibt sich

$$y_n = \sum_{j=0}^{N'-1} w^{2nj} x_{2j} + \sum_{j=0}^{N'-1} w^{(2j+1)n} x_{2j+1}$$

und folglich

$$y_n = \sum_{j=0}^{N'-1} w'^{nj} x_{2j} + w^n \sum_{j=0}^{N'-1} w'^{nj} x_{2j+1} =: y_n^0 + w^n y_n^1. \quad (2)$$

Ersetzen wir hier  $n$  durch  $n + N'$ , so erhalten wir wegen

$$w^{N'} = w^{N/2} = -1, \quad w'^{N'} = 1$$

die weiteren Formeln

$$y_{n+N'} = \sum_{j=0}^{N'-1} w'^{nj} x_{2j} - w^n \sum_{j=0}^{N'-1} w'^{nj} x_{2j+1} = y_n^0 - w^n y_n^1 \quad (3)$$

mit denselben Summen wie in (2). Diese Summen  $y_n^0$  und  $y_n^1$  sind nichts anderes als die  $N'$  Komponenten der Vektoren  $\mathbf{W}'\mathbf{x}^0$  und  $\mathbf{W}'\mathbf{x}^1$ , wobei wir zur Abkürzung

$$\mathbf{x}^0 := \begin{bmatrix} x_0 \\ x_2 \\ \vdots \\ x_{N-2} \end{bmatrix}, \quad \mathbf{x}^1 := \begin{bmatrix} x_1 \\ x_3 \\ \vdots \\ x_{N-1} \end{bmatrix}$$

gesetzt haben. Schreiben wir nun die Gleichungen (2) für  $0 \leq n \leq N' - 1$  untereinander und darunter die Gleichungen (3) für  $0 \leq n \leq N' - 1$ , so erhalten wir die folgende Matrixgleichung, die den Vektor  $\mathbf{y} = \mathbf{W}\mathbf{x}$  auf die Vektoren  $\mathbf{W}'\mathbf{x}^0$  und  $\mathbf{W}'\mathbf{x}^1$  der halben Länge zurückführt:

(7.6)

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_{N-1} \end{bmatrix}^{\triangleleft} = \begin{bmatrix} 1 & & & & & & & 1 \\ & 1 & & & & & & w \\ & & \ddots & & & & & \ddots \\ & & & 1 & & & & w^{N'-1} \\ 1 & & & & -1 & & & \\ & 1 & & & & -w & & \\ & & \ddots & & & & \ddots & \\ & & & 1 & & & & -w^{N'-1} \end{bmatrix} \begin{bmatrix} x_0 \\ x_2 \\ \vdots \\ x_{N-2} \end{bmatrix}^{\triangleleft} + \begin{bmatrix} \\ \\ \\ \\ \\ \\ \\ \\ \end{bmatrix} \begin{bmatrix} x_1 \\ x_3 \\ \vdots \\ x_{N-1} \end{bmatrix}^{\triangleleft}.$$

Die hier erschienene  $(N \times N)$ -Matrix

$$\mathbf{B} := \mathbf{B}_N := \begin{bmatrix} \text{diag}(1, 1, \dots, 1) & \text{diag}(1, w, \dots, w^{N'-1}) \\ \text{diag}(1, 1, \dots, 1) & -\text{diag}(1, w, \dots, w^{N'-1}) \end{bmatrix}, \quad w := e^{2\pi i/N},$$

wird **Schmetterlingsmatrix** genannt, da sie die jeweiligen Daten parallel und gleichzeitig übers Kreuz an die nächste Stufe weitergibt (siehe die Fig. 7.3.2).

Bevor wir den in (7.6) beschriebenen Vorgang iterieren, wollen wir berechnen, wieviele Multiplikationen die vollständige DFT eines Datensatzes der Länge  $N = 2^r$  erfordert. Es bezeichne  $\mu(r)$  diese Anzahl. Man hat offensichtlich  $\mu(0) = 0$ , und an (7.6) mit  $N' = 2^r$  liest man die Abschätzung

$$\mu(r + 1) \leq 2^r + 2\mu(r)$$

ab. Hieraus ergibt sich mit vollständiger Induktion die Ungleichung

$$\mu(r) \leq r 2^{r-1} \quad (r \geq 1)$$

oder eben, dass für die DFT von Datensätzen der Länge  $N = 2^r$  insgesamt weniger als  $0.5 N \log_2 N$  Multiplikationen erforderlich sind. Die weitere Inspektion von (7.6) erweist, dass die Anzahl der Additionen etwa doppelt so gross ist.

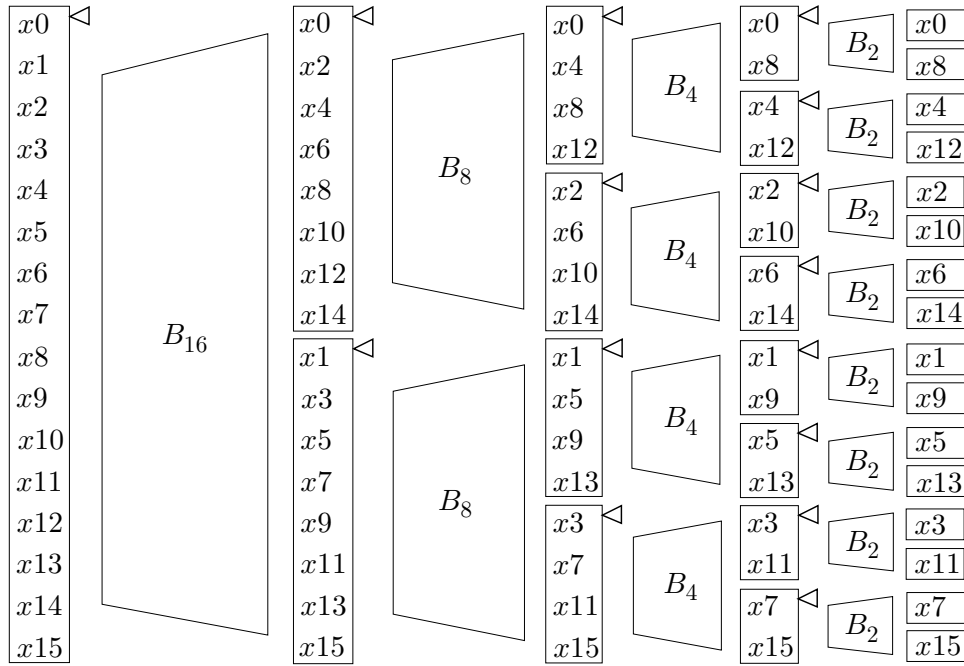


Fig. 7.3.1

Die Figur 7.3.1 zeigt am Beispiel  $N = 16$ , welche Transformatierten von Teildatensätzen sich einstellen, wenn das Prinzip (7.6) rekursiv angewandt wird, bis  $N' = 1$  ist. Zuäusserst rechts erscheinen die Ausgangsdaten  $x_k$  in einer eigentümlichen Reihenfolge. Was hat es damit für eine Bewandnis? Um dahinter zu kommen, schreiben wir die vorgefundene Nummernfolge 0, 8, 4, 12, ..., 7, 15 in Binärdarstellung auf:

0000, 1000, 0100, 1100, 0010, 1010, 0110, 1110,  
 0001, 1001, 0101, 1101, 0011, 1011, 0111, 1111 .

Werden diese 16 Bitstrings nacheinander “von hinten nach vorne” gelesen, so erhält man

0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111,  
1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111,

also die übliche lexikographische Anordnung. Die rechts in der Fig. 7.3.1 erschienene Reihenfolge der  $x_k$  entsteht also durch lexikographische Anordnung ihrer Binäradressen, wobei aber diese Adressen von rechts nach links gelesen werden. Der resultierende Umordnungsprozess wird im Jargon **bit reversal** genannt.

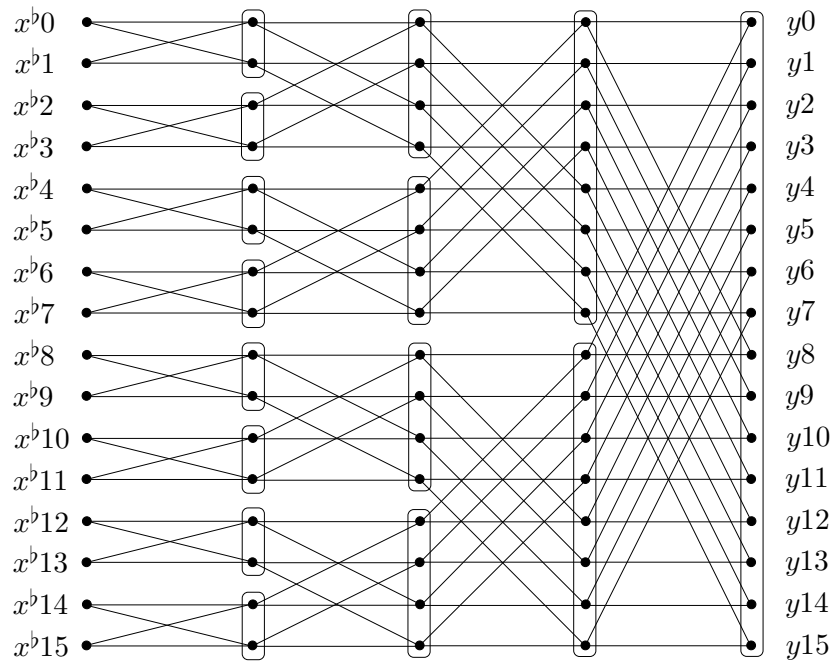


Fig. 7.3.2

Damit haben wir alles beisammen, und wir können definitiv erklären, wie die schnelle Fourier-Transformation für einen Datenvektor  $\mathbf{x}$  der Länge  $N = 2^r$  funktioniert. Hierzu müssen wir die Fig. 7.3.1 von rechts nach links lesen. Um den Kolonnenvektor  $\mathbf{x}$  mit der DFT-Matrix  $\mathbf{W}$  zu multiplizieren, hat man danach folgendes zu tun:

1. Die Komponenten  $x_k$  von  $\mathbf{x}$  sind durch *bit reversal* umzuordnen; es entsteht der Datensatz  $\mathbf{x}^b = (x_0^b, x_1^b, \dots, x_{N-1}^b)$ , bestehend aus  $N$  Einzeldaten.
2. Nun werden zunächst je zwei aufeinanderfolgende Einzeldaten mit Hilfe von  $\mathbf{B}_2$  zu einem Paar verknüpft; dann je zwei aufeinanderfolgende Paare mit Hilfe von  $\mathbf{B}_4$  zu einem Vektor der Länge 4, undso weiter. Im letzten Schritt werden zwei Vektoren der Länge  $N' = 2^{r-1}$  mit Hilfe von  $\mathbf{B}_N$  zum gewünschten Output  $\mathbf{y}$  verknüpft.

Der im zweiten Schritt resultierende Datenfluss ist in Fig. 7.3.2 (von links nach rechts zu lesen!) dargestellt. Aus (2) und (3) geht hervor, dass jeder neu berechnete Wert nur von zwei unmittelbar vorangehenden Werten direkt abhängt.

Der Normalverbraucher braucht sich natürlich um alle diese Details nicht zu kümmern, da die FFT in den meisten handelsüblichen Programmpaketen einprogrammiert ist.